Omne: Cadence-Separated Consensus for Dual-Finality Blockchain Infrastructure

Greg B — Omne Foundation (OmneDAO)

March 2026

Abstract

We introduce Cadence-Separated Consensus (CSC), a consensus framework in which distinct block-production cadences operate on a shared validator set with cryptographic binding between layers. We prove that in a CSC system where the security cadence is k times the commerce cadence, any transaction confirmed at the commerce layer achieves settlement finality within one security interval, and the security overhead per commerce transaction is O(1/k). We conjecture that a dual-token system pairing a deflationary staking token with an inflationary fee-recycled commerce token converges to a stable supply equilibrium under bounded network conditions. We prove a quantitative lower bound on peg duration as a function of treasury reserves and demand volatility (Theorem 2), and conjecture that microscopic transaction fees are structurally sustainable—not temporarily subsidized—when block rewards, fee recycling, and compute cross-subsidization jointly exceed validator operating costs independent of fee revenue (Conjecture 2). We present Omne, a layer-one blockchain that instantiates CSC with a 3-second commerce cadence (k = 180), producing a 3-second commerce layer for point-of-sale finality and a 9-minute security layer for settlement-grade guarantees, unified through Proof of Validated Execution and Resource Attestation (PoVERA). The protocol employs a dual-token model: OMC, an inflationary commerce token with fee recycling economics (8% anti-spam burn, 92% validator recycling) and 18-decimal (quar) precision enabling sub-cent transaction costs, and OGT, a deflationary governance token with a 21-million-unit hard cap governed by a fixed-supply halving schedule. Transaction fees are recycled rather than aggressively burned, preserving OMC's soft dollar peg by avoiding the deflationary pressure that aggressive fee burning would impose on a pegged commerce token. A native smart contract pipeline executes WASM bytecode deterministically via the Axiom Runtime across heterogeneous hardware, with SHA-256 state commitments embedded in every security block. The Omne Orchestration Network (OON) extends the validator infrastructure into an off-chain distributed compute marketplace, producing cryptographic attestation certificates and cross-subsidizing on-chain transaction fees at a rate of 30% of compute revenue. The Omne Media Protocol (OMP) provides decentralized file storage through a hybrid architecture: asset manifests, SHA-256 merkle roots, and storage proofs live on-chain in consensus state, while raw chunk data is distributed across dedicated storage nodes with configurable redundancy and erasure coding. Anti-spam protections rely exclusively on non-fee controls—rate limiting, stake-weighted quality-of-service, throttling, and circuit breakers—ensuring that microscopic fees remain a permanent architectural property rather than a temporary subsidy. The CSC framework is general: other instantiations may adopt different cadence ratios, consensus mechanisms, and token models. We describe the theoretical framework, its Omne instantiation, and the complete protocol in sufficient detail for independent implementation and formal analysis.

Table of Contents

1 Introduction

The promise of blockchain-based commerce has been constrained by a fundamental tension between the properties that make distributed ledgers trustworthy and the requirements that make payment systems usable. A merchant processing a point-of-sale transaction requires confirmation within seconds. A developer deploying a token contract should not pay more in gas than the contract's first hundred users will transact. A small business owner should not watch their $50 transfer lose $4.70 to fee volatility between mempool submission and block inclusion.

Existing layer-one protocols have optimized for one end of this spectrum at the expense of the other. High-throughput chains achieve low latency by weakening finality guarantees, creating reorg risk that is unacceptable for commerce. Security-first chains achieve strong finality at the cost of confirmation times measured in minutes and fee markets that auction block space to the highest bidder. No production chain has resolved what we term the *latency-security-fee trilemma*: the simultaneous achievement of sub-second commercial finality, settlement-grade security, and deterministically microscopic fees.

We propose that this trilemma admits a theoretical resolution through a framework we call *Cadence-Separated Consensus* (CSC): a consensus architecture in which distinct block-production cadences operate on a shared validator set with cryptographic binding between

layers. Rather than forcing a single block cadence to serve both commerce and settlement, CSC defines independent cadences—each optimized for its purpose—and binds them through a periodic cryptographic commitment that inherits the stronger layer's finality guarantee. The three architectural decisions at the core of Omne are not ad hoc design choices but interlocking consequences of the CSC framework:

1. **Dual-cadence block production.** A 3-second commerce cadence ($\tau_c = 3$s) for point-of-sale finality and a 9-minute security cadence ($\tau_s = 540$s) for settlement-grade guarantees, with $k = 180$ commerce blocks rolling into each security commitment.
2. **Dual-token economics.** A deflationary governance token (OGT) and an inflationary commerce token (OMC) whose supply dynamics are structurally aligned with the properties each cadence layer requires—scarcity and long-term alignment for security, abundance and price stability for commerce.
3. **Non-fee anti-spam.** Rate limiting, stake-weighted quality-of-service, throttling, and circuit breakers replace fee escalation as the spam deterrent, allowing transaction fees to be set at the cost floor rather than the market-clearing price.

This paper makes five contributions:

1. We define Cadence-Separated Consensus formally and prove that transactions in a CSC system achieve settlement finality within τ_s with security overhead $O(1/k)$ per commerce transaction (Section 2, Theorem 1).
2. We state the *Dual Equilibrium Conjecture*: that a paired deflationary-inflationary token model with fee recycling converges to a stable supply equilibrium under bounded network conditions (Section 2, Conjecture 1).
3. We prove a quantitative lower bound on peg stability duration as a function of treasury reserves and demand volatility, modeling the OMC band peg as a Lyapunov-stable control system (Section 2, Theorem 2).
4. We conjecture *Structural Fee Sustainability*: that microscopic fees are a permanent architectural property when block rewards, fee recycling, and compute cross-subsidization jointly exceed validator operating costs independent of fee revenue (Section 2, Conjecture 2).
5. We describe a complete reference implementation—Omne—including its consensus mechanism, token economics, fee model, execution environment, smart contract system, compute orchestration layer, decentralized storage protocol, and security model, in sufficient detail for independent implementation and economic analysis (Sections 3–14).

We further hypothesize that compute orchestration and decentralized storage can be integrated at the protocol level, producing utilization-driven revenue streams that reinforce fee sustainability without introducing external dependencies.

2 Cadence-Separated Consensus

This section defines Cadence-Separated Consensus as a general theoretical framework, independent of any specific instantiation. The definitions and results are parameterized by the cadence ratio, the consensus mechanism at each layer, and the token model. Omne's particular parameter choices are described beginning in Section 3.

2.1 Definition

Definition 1 (Cadence-Separated Consensus). A *cadence-separated consensus system* is a tuple $(V, \tau_c, \tau_s, \Pi_c, \Pi_s, \mathcal{B})$ where:

- V is a shared validator set of n nodes.
- τ_c is the *commerce cadence*: the block production interval for the fast layer.
- $\tau_s = k \cdot \tau_c$ is the *security cadence*: the block production interval for the settlement layer, where $k \in \mathbb{N}$, $k \geq 2$.
- Π_c is the consensus protocol governing commerce block production.
- Π_s is the consensus protocol governing security block production, implementing BFT finality with quorum threshold q.
- $\mathcal{B} : \{b_1, \ldots, b_k\} \to C$ is a *binding function* that maps a sequence of k commerce blocks to a cryptographic commitment C embedded in the subsequent security block.

The key structural property is that V participates in *both* cadences simultaneously. This distinguishes CSC from multi-chain architectures (which use separate validator sets and require bridges) and from sharding (which partitions the validator set across execution domains). In a CSC system, every validator attests to both the commerce execution history and the security settlement state.

The cadence ratio $k = \tau_s/\tau_c$ is the fundamental design parameter. It governs the trade-off between settlement latency and security amortization: larger k amortizes BFT overhead across more commerce transactions but increases the worst-case settlement latency. The choice of k is an engineering decision, not a theoretical constraint; Theorem 1 holds for all $k \geq 2$.

2.2 Binding Theorem

Theorem 1 (Settlement Finality and Security Amortization). In a cadence-separated consensus system $(V, \tau_c, \tau_s, \Pi_c, \Pi_s, \mathcal{B})$ where $\tau_s = k \cdot \tau_c$ and Π_s implements BFT finality with quorum threshold $q \geq \lfloor 2n/3 \rfloor + 1$:

(a) Any transaction t confirmed in commerce block b_i achieves settlement finality no later than security block S_j where $j = \lceil i/k \rceil$. The maximum settlement latency is τ_s.

(b) The security overhead per commerce transaction is $O(1/k)$.

Proof sketch.

(a) By construction, security block S_j contains the binding commitment $\mathcal{B}(b_{(j-1)k+1}, \ldots, b_{jk})$, which is a cryptographic digest (in Omne's instantiation, a SHA-256 Merkle root) over the commerce blocks in the j-th epoch. BFT finality of S_j requires attestation from $q \geq \lfloor 2n/3 \rfloor + 1$ validators, each of which independently computes $\mathcal{B}$ by replaying the k commerce blocks and verifying the commitment. Once S_j achieves BFT finality, the commitment is irreversible under the standard BFT assumption (fewer than $n/3$ Byzantine validators [8]). Since t is included in b_i and b_i is covered by $\mathcal{B}$, transaction t inherits the security layer's finality guarantee. The worst case occurs when t is in the first commerce block of an epoch ($i = (j-1)k + 1$), yielding settlement latency τ_s. The best case occurs when t is in the last commerce block ($i = jk$), yielding settlement latency approaching τ_c.

(b) Π_s produces one BFT round per k commerce blocks. For a HotStuff-derived linear BFT protocol [6], the message complexity per round is $O(n)$. This cost is amortized across k commerce

blocks, yielding $O(n/k)$ messages per commerce block. For a fixed validator set size n, the per-transaction security overhead is $O(1/k)$. For classical BFT protocols [8] with $O(n^2)$ message complexity, the per-transaction overhead becomes $O(n/k)$, still decreasing linearly in k.

Corollary 1. In Omne's instantiation ($k = 180$, linear BFT), the security overhead per commerce transaction is $1/180$ of the cost of a single BFT round. This amortization is what allows the commerce layer to operate at a 3-second cadence without imposing BFT message exchange on every block.

2.3 Dual Equilibrium Conjecture

Conjecture 1 (Dual Token Equilibrium). Let a CSC system employ a dual-token model consisting of:

- A *governance token* G with hard supply cap S_G, a halving block reward schedule with permanent floor r_G^{min}, and a staking requirement $\sigma \in [\sigma_{\text{min}}, \sigma_{\text{max}}]$ per validator.
- A *commerce token* M with no supply cap, a block reward R_M (subject to its own halving schedule), a fee recycling rate $(1 - \beta)$ where β is the anti-spam burn rate, and an external price stabilization mechanism bounding market price within $[p_{\text{floor}}, p_{\text{ceiling}}]$.

If the following conditions hold:

1. R_M is small relative to circulating supply M_{circ},
2. β is small ($\beta \ll 1$),
3. Network transaction volume $V(t)$ is bounded above by a constant V_{max},
4. The price stabilization mechanism is solvent (treasury reserves exceed expected buy-back demand within any governance-defined epoch),

then the system converges to an equilibrium in which:

- G's liquid supply decreases monotonically toward a minimum determined by staking demand and infrastructure bonding requirements.
- M's circulating supply oscillates within a bounded range determined by R_M, $V(t)$, and β.
- The net supply change per security block satisfies $|\Delta M| < \varepsilon$ for a bound ε that decreases as network utilization matures.

This remains a conjecture because proving convergence requires assumptions about market participant behavior—specifically, the rationality and responsiveness of treasury governance—that cannot be derived from protocol mechanics alone. We invite formal economic analysis and agent-based simulation to explore the conjecture's boundary conditions.

2.4 Separability Property

Property 1 (Independent Verifiability). In a cadence-separated consensus system, correctness decomposes into three independently verifiable components:

1. **Commerce layer correctness.** Each commerce block contains valid transactions, correct state transitions, and proper fee accounting. This is verifiable by replaying the block in isolation against the prior state.
2. **Security layer correctness.** Each security block contains a valid BFT quorum certificate, correct reward and slash accounting, and well-formed system operations. This is verifiable

from the security block's internal structure and the validator registry.

3. **Cross-layer binding correctness.** The commitment $\mathcal{B}$ in security block S_j matches the independently computed commitment from replaying commerce blocks $b_{(j-1)k+1}, \ldots, b_{jk}$.

A system is *CSC-correct* if and only if all three components are correct. This decomposition simplifies formal verification: each component can be specified and verified in isolation, and the composition is secured by the cryptographic binding function $\mathcal{B}$. In proof-assistant frameworks, the three components correspond to three independent proof obligations, reducing the verification surface compared to a monolithic consensus specification.

2.5 Generality

CSC is a *framework*, not a protocol specification. The definitions and results above are parameterized by τ_c, τ_s, k, Π_c, Π_s, and $\mathcal{B}$. Omne instantiates CSC with $\tau_c = 3$s, $k = 180$, and HotStuff-derived linear BFT at the security layer. Other instantiations are possible:

- A high-frequency trading chain might set $\tau_c = 200$ms with $k = 3{,}000$, achieving sub-second commerce confirmation and 10-minute settlement.
- A settlement-focused network might set $\tau_c = 30$s with $k = 20$, trading commerce speed for a simpler security schedule.
- A multi-cadence extension could introduce a third layer with cadence $\tau_a = m \cdot \tau_s$ for archival or governance finality, though we do not explore this generalization here.

The Binding Theorem holds for any choice of parameters satisfying the BFT quorum condition. The Dual Equilibrium Conjecture is independent of cadence parameters and applies to any dual-token system satisfying conditions (1)–(4). The key insight is that cadence separation is a *design dimension* that existing consensus literature has not explored as a first-class primitive. Lamport's foundational ordering framework [7] establishes that logical time in distributed systems is defined by causal ordering, not clock synchrony; CSC extends this by observing that *different classes of events may require different temporal granularities* within a single system, and that these granularities can be composed rather than compromised.

2.6 Structural Fee Sustainability

Conjecture 2 (Structural Fee Sustainability). In a CSC system with dual-token economics, define the *per-epoch validator revenue function*:

$$\rho(t) \;=\; \frac{R_M(t) \;+\; (1-\beta)\,F(t) \;+\; \gamma\,\Omega(t)}{n}$$

where $R_M(t)$ is the OMC block reward at security epoch t (subject to halving with permanent floor $R_M^{\min} > 0$), β is the anti-spam burn rate, $F(t)$ is total transaction fees collected per epoch, γ is the compute cross-subsidy rate, $\Omega(t)$ is Omne Orchestration Network compute revenue per epoch, and $n = |V|$ is the number of active validators.

If:

1. The block reward has a permanent floor: $R_M(t) \geq R_M^{\min} > 0$ for all t.
2. Compute utilization provides a positive revenue floor: $\Omega(t) \geq \Omega_{\min} > 0$.
3. The *sustainability condition* holds:

$$R_M^{\min} + \gamma \cdot \Omega_{\min} \;>\; n \cdot c_v$$

where c_v is the per-validator operating cost per security epoch,

then $\rho(t) > c_v$ for all t, and microscopic transaction fees are structurally sustainable—not temporarily subsidized.

Observation. Condition (3) is independent of $F(t)$. Fee revenue enters as surplus above break-even, not as a structural dependency. This permits transaction fees to be set at the commerce cost floor because validator economics are sustained by block rewards and compute revenue even when $F(t) \to 0$. As $R_M(t)$ halves toward $R_M^{\min}$, the compute cross-subsidy $\gamma \cdot \Omega(t)$ becomes the dominant margin above break-even, producing an organic transition from issuance-funded to utilization-funded validator economics without requiring fee increases. In Omne's instantiation ($\beta = 0.08$, $\gamma = 0.30$, $n \leq 100$), this structure is designed to hold from genesis.

This remains a conjecture because condition (2)—a positive compute revenue floor—depends on exogenous market demand for OON services that cannot be guaranteed by protocol mechanics alone.

2.7 Peg Stability

Model. Let time be discrete, indexed by blocks ($t \in \mathbb{N}$). Let $p_t \in \mathbb{R}_{>0}$ denote the OMC market price in a reference currency. Fix floor price p_f, ceiling price p_c, target midpoint $p^* = (p_f + p_c)/2$, and band width $w = p_c - p_f$. Let T_t denote treasury reserves (reference currency). Define:

- δ: demand volatility bound (maximum net demand deviation from mean, in demand units per block).
- Δp_{max}: price impact per unit demand (reference currency per demand unit).

The exogenous per-block price shock is $\xi_t = \Delta p_{\text{max}} \cdot (D_t - \bar{D})$, satisfying $|\xi_t| \leq \delta \cdot \Delta p_{\text{max}}$.

Assumptions.

(A1) *Bounded shocks.* $|\xi_t| \leq \delta \cdot \Delta p_{\text{max}}$ for all t.

(A2) *Treasury depletion.* Each intervention costs the treasury $|u_t|$ reference-currency units, where u_t is the price correction applied at step t. Treasury dynamics: $T_{t+1} = T_t - |u_t|$.

(A3) *Band-clamping controller.* The treasury implements:

$$p_{t+1} = \begin{cases} \text{proj}_{[p_f,\, p_c]}(p_t + \xi_t) & \text{if } T_t > 0 \\ p_t + \xi_t & \text{if } T_t = 0 \end{cases}$$

where $\text{proj}_{[a,b]}(x) = \min(\max(x, a), b)$, and the control input is $u_t = \text{proj}_{[p_f,p_c]}(p_t + \xi_t) - (p_t + \xi_t)$. This is the mathematical representation of the faucet (sell OMC at ceiling) and buy-back (purchase OMC at floor) mechanism: the treasury intervenes with exactly the magnitude required to clamp the price within the band.

Theorem 2 (Peg Stability Duration). Under (A1)–(A3), with initial treasury $T_0 > 0$ and $p_0 \in [p_f, p_c]$:

(a) Finite-horizon bound. The price satisfies $p_t \in [p_f, p_c]$ for all $t \leq t^*$, where

$$t^* = \left\lfloor \frac{T_0}{\delta \cdot \Delta p_{\max}} \right\rfloor.$$

(b) Lyapunov dissipation. Define $L(p) = (p - p^*)^2$ (we reserve V for the validator set of Definition 1). Along controlled trajectories ($T_t > 0$):

- $L(p_t) \leq (w/2)^2$ for all $t \leq t^*$ (band invariance implies a Lyapunov bound).
- At every step where the controller activates ($u_t \neq 0$), $L(p_{t+1}) < L(p_t + \xi_t)$: the controller strictly dissipates deviation energy relative to the uncontrolled trajectory.

(c) Indefinite stability under mean-reversion. If $\mathbb{E}[\xi_t] = 0$ and $|\xi_t| \leq \delta \cdot \Delta p_{\max}$, the peg is maintained indefinitely provided treasury reserves remain above a critical threshold:

$$T_t > T_{\min} = \delta \cdot \Delta p_{\max} \cdot \tau_{\text{epoch}}$$

where τ_{epoch} is the governance response time (blocks between treasury replenishment decisions).

Proof.

(a) By induction. Base case: $p_0 \in [p_f, p_c]$ by hypothesis. Inductive step: suppose $p_t \in [p_f, p_c]$ and $T_t > 0$. Then $p_{t+1} = \text{proj}_{[p_f, p_c]}(p_t + \xi_t) \in [p_f, p_c]$ by (A3). The intervention cost is $|u_t| = |p_t + \xi_t - \text{proj}_{[p_f, p_c]}(p_t + \xi_t)|$. Since $p_t \in [p_f, p_c]$ and $|\xi_t| \leq \delta \cdot \Delta p_{\max}$, the overshoot beyond the band is at most $\delta \cdot \Delta p_{\max}$, so $|u_t| \leq \delta \cdot \Delta p_{\max}$. Therefore $T_{t+1} \geq T_t - \delta \cdot \Delta p_{\max}$. Summing: $T_t \geq T_0 - t \cdot \delta \cdot \Delta p_{\max} > 0$ for all $t < T_0/(\delta \cdot \Delta p_{\max})$. □

(b) The bound $L(p_t) \leq (w/2)^2$ follows immediately from $p_t \in [p_f, p_c]$, since $|p_t - p^*| \leq w/2$. For the dissipation claim, suppose $u_t \neq 0$, i.e., $p_t + \xi_t \notin [p_f, p_c]$. Without loss of generality, assume $p_t + \xi_t > p_c$ (the floor case is symmetric). Then $p_t + \xi_t - p^* > p_c - p^* = w/2$ and $p_{t+1} = p_c$, so $p_{t+1} - p^* = w/2 < p_t + \xi_t - p^*$. Therefore $L(p_{t+1}) = (w/2)^2 < (p_t + \xi_t - p^*)^2 = L(p_t + \xi_t)$. □

(c) Under mean-zero shocks, the expected treasury drain per step is $\mathbb{E}[|u_t|] \leq \delta \cdot \Delta p_{\max} \cdot \Pr[p_t + \xi_t \notin [p_f, p_c]]$. Over a governance epoch of τ_{epoch} blocks, the worst-case cumulative drain (every block triggers intervention at maximum cost) is $\delta \cdot \Delta p_{\max} \cdot \tau_{\text{epoch}} = T_{\min}$. If governance replenishes the treasury at each epoch boundary and $T_t > T_{\min}$ at each replenishment, the treasury cannot reach zero between replenishments, maintaining the peg indefinitely. Under mean-zero demand, the actual drain is strictly less than $T_{\min}$ per epoch (the price spends time in the band interior where $u_t = 0$), so $T_{\min}$ is a conservative upper bound. □

Remark. Theorem 2 transforms the informal claim "the treasury will maintain the peg" into a falsifiable prediction. The bound t^* is a function of three measurable quantities: initial treasury reserves T_0, demand volatility δ, and per-unit price impact $\Delta p_{\max}$. Governance can monitor these quantities in real time and take corrective action (treasury replenishment, band adjustment) before the bound is approached. The critical threshold $T_{\min}$ gives a concrete early-warning level: if reserves fall below $T_{\min}$, the peg is at risk within the next governance epoch.

2.8 Layered Duality

Design Principle 1 (Layered Duality). In a cadence-separated consensus system, each consensus layer pairs with a token whose supply dynamics mirror the layer's temporal and economic character:

Layer	Properties	Paired token	Supply dynamics
Commerce (τ_c)	Fast, abundant, frequent	M (OMC)	Inflationary, fee-recycled, peg-targeted
Security (τ_s)	Slow, scarce, final	G (OGT)	Deflationary, halving, hard-capped

Cadence separation and dual-token design are not independent architectural choices. The commerce layer requires a plentiful, price-stable medium of exchange to support microscopic fees; the security layer requires a scarce, long-term-aligned staking asset to incentivize honest validation. A single token serving both roles forces contradictory supply pressures—the conflict identified in Section 6.1. Layered Duality asserts that the *natural* token model for a CSC system assigns one token per cadence layer, with supply mechanics that are structurally coupled to the layer's operational requirements. Conjecture 1 (dual token equilibrium), Conjecture 2 (structural fee sustainability), and the quantitative peg stability bound (Theorem 2) are consequences of this coupling: all three results depend on the alignment between token supply dynamics and layer properties, not on arbitrary parameter choices.

3 Architecture Overview

Omne instantiates the CSC framework (Section 2) as a single logical chain with two block-production cadences operating in parallel. The commerce layer produces blocks every 3 seconds ($\tau_c = 3$s), targeting point-of-sale and e-commerce confirmation latency. The security layer produces blocks every 540 seconds ($\tau_s = 540$s, 9 minutes), each embedding a cryptographic commitment over the 180 commerce blocks produced during that interval ($k = 180$). Validators participate in both layers through a unified Proof of Validated Execution and Resource Attestation (PoVERA) consensus mechanism.

Transaction execution occurs in the Axiom Runtime, a WASM-based deterministic execution environment that guarantees identical state transitions across heterogeneous hardware. Every security block contains an `ExecutionStateCommitment` comprising a SHA-256 state root, an execution trace hash, and a per-tier gas budget report, binding the execution history of the preceding commerce epoch to the security chain.

The Omne Orchestration Network (OON) extends the validator infrastructure into an off-chain compute marketplace. Registered nodes rent idle capacity for workloads such as AI training, 3D rendering, and data processing. Compute revenue cross-subsidizes on-chain transaction fees at a protocol-defined rate of 3,000 basis points (30%), creating a sustainable funding mechanism for microscopic fees without relying on inflationary pressure.

The Omne Media Protocol (OMP) extends the network with decentralized file storage. Files are serialized into 256 KB chunks, content-addressed via SHA-256, and distributed across dedicated

storage nodes with configurable redundancy (2–7× replication) and optional Reed-Solomon erasure coding. Storage nodes bond 500 OGT, build reputation through proof-of-storage challenges, and earn fees via escrow-based periodic settlement. Asset manifests and merkle roots are maintained on-chain; raw chunk data lives off-chain on storage nodes. This hybrid architecture avoids bloating consensus state while providing on-chain integrity guarantees for all stored data.

4 Dual-Layer Consensus

This section details the concrete block structures and production rules that instantiate the CSC framework (Section 2) in the Omne protocol.

4.1 Commerce Layer

The commerce layer produces blocks at a fixed cadence of 3 seconds (`COMMERCE_BLOCK_TIME_SECS = 3`). Each commerce block contains transactions prioritized by a three-tier system: Commerce priority (weight 3), Standard priority (weight 2), and Compute priority (weight 1). Block execution budgets allocate 70% of available time to FastVM transactions (commerce), 20% to StandardVM, and 10% to ComputeVM.

Commerce blocks achieve probabilistic finality upon production and deterministic finality upon inclusion in a security block. For point-of-sale use cases, the 3-second probabilistic finality is sufficient; the merchant receives confirmation before the customer leaves the counter. Per Theorem 1, full settlement finality is achieved when the enclosing security block reaches BFT quorum.

4.2 Security Layer

The security layer produces blocks every 540 seconds (9 minutes), with each security block spanning exactly 180 commerce blocks:

$$\frac{540 \text{ s}}{3 \text{ s/block}} = 180 \text{ commerce blocks per security block}$$

Each security block contains:

Field	Description
`commerce_state_commitment`	Merkle commitment over 180 commerce blocks
`execution_state_commitment`	SHA-256 state root, execution trace hash, budget report hash
`validator_rewards`	Per-validator reward breakdown for the epoch
`block_fees`	Aggregated fee accounting (collected, burned, distributed)
`slashed_stake_ogt`	Total OGT slashed during the epoch
`system_operations`	Structured audit trail of all protocol-level token operations (reward mints, fee distributions, vault seeds, slash burns) with from/to addresses, amounts, and token type
`proposer_signature`	Ed25519 signature over the block hash by the block producer
`proposer_public_key`	32-byte Ed25519 public key of the signing validator

Field	Description
`vdp_commitments`	Verifiable-Deterministic-Parallelizable attestations
`computational_revenue_quar`	OON compute revenue for cross-subsidization
`storage_fee_revenue_quar`	OMP storage fee revenue for the epoch
`commit_qc`	Commit-phase quorum certificate providing cryptographic proof that a supermajority of validators attested to the block's validity
`bft_finalized`	Boolean indicating whether the block has achieved BFT finality through the three-phase commit protocol

The 9-minute cadence was chosen to balance two constraints: the interval must be long enough to amortize the cost of BFT finality across many commerce transactions (the $O(1/k)$ amortization established in Theorem 1), and short enough that the security chain provides meaningful settlement guarantees within a time frame comparable to the ~10-minute cadence characteristic of security-first proof-of-work chains.

4.3 Commerce-to-Security Rollup

At the close of each 180-block commerce epoch, the block producer constructs an `ExecutionStateCommitment`:

```
ExecutionStateCommitment {
    state_root:           SHA-256(state trie),
    execution_trace_hash: SHA-256(deterministic execution trace),
    budget_report_hash:   SHA-256(per-tier gas budget report),
    source_block_height:  commerce block height
}
```

This commitment is the concrete realization of the binding function $\mathcal{B}$ from Definition 1. It is embedded in the subsequent security block, creating an immutable binding between the commerce execution history and the security chain. Any validator can independently reconstruct the commitment by replaying the 180 commerce blocks, providing a deterministic verification path—the mechanism through which Property 1 (Independent Verifiability) is enforced.

4.4 System Operations

Each security block embeds a structured `system_operations` array that records every protocol-level token mutation performed during block production. This provides an Etherscan-equivalent audit trail for operations that are not user-initiated transactions but nonetheless alter account balances:

Operation Type	Description
`RewardMint`	OGT or OMC minted to a validator as a block reward
`FeeDistribution`	Fee revenue routed from the fee vault to treasury, validator pool, or slash sink

Operation Type	Description
`FeeVaultSeed`	Accumulated commerce-layer fees transferred into the fee vault for distribution

Each operation records the `from` address, `to` address, `amount` (in base units: quar for OMC, micro-OGT for OGT), and `token` type. Explorers and auditors can reconstruct the complete economic history of the chain from these records without relying on off-chain indexers or event logs.

System operations are not transactions — they carry no sender signature, nonce, or gas cost, and they do not occupy mempool space. They are deterministic consequences of the reward schedule and fee split configuration. Their state mutations are committed to the Merkle tree before the security block's `execution_state_commitment.state_root` is read, ensuring that the state root covers all economic activity in the block.

When a validator receives a proposed security block, it independently re-executes the embedded system operations and verifies six accounting invariants: vault seed balance consistency, fee distribution total accuracy, OGT mint total correctness, OMC mint total correctness, absence of zero-amount operations, and validator reward coverage completeness. A block whose system operations violate any invariant is rejected before the validator casts a BFT attestation vote, preventing blocks with inconsistent economic state from reaching finality.

5 PoVERA Consensus

Proof of Validated Execution and Resource Attestation (PoVERA) is Omne's consensus mechanism, implementing the CSC framework's Π_c and Π_s protocols within a single validator set. It combines four components:

1. **Proof of Stake (PoS):** Validators bond OGT within the dynamic stake range of 15–28 OGT (`MIN_VALIDATOR_STAKE = 15`, `MAX_VALIDATOR_STAKE = 28`). The active validator set is capped at 100 nodes. Delegators may participate with a minimum of 1 OGT.
2. **Proof of Valuable Computation (PoVC):** Validators that contribute compute capacity through the Omne Orchestration Network earn attestation certificates, which factor into block producer selection. This creates a direct incentive for validators to make surplus capacity available for off-chain workloads.
3. **RANDAO Randomness:** Block producer selection uses a RANDAO-derived beacon, weighted by stake, to ensure pseudorandom and manipulation-resistant leader election.
4. **BFT Finality:** The security layer implements a HotStuff-derived [6] three-phase BFT engine (Prepare PreCommit Commit) requiring a $(2n/3) + 1$ supermajority quorum at each phase. Double-vote detection is enforced by SafetyRules, which track the highest voted round per validator and reject equivocating messages. Each vote is constructed as SHA-256$(\text{block_hash} \| \text{round} \| \text{phase})$ and signed with the validator's Ed25519 consensus key. BFT votes propagate via a dedicated `BftVote` P2P message type at Critical priority, ensuring consensus traffic is never shed under load. Quorum certificates are produced at each phase transition and persisted; the commit-phase quorum certificate is embedded

in the finalized `SecurityBlock`, providing the cryptographic proof of supermajority attestation required by Theorem 1. On devnet with a single validator, security blocks self-finalize through all three phases in a single pass. On multi-validator networks, vote exchange occurs via P2P and blocks finalize when the commit phase reaches supermajority.

Validators are rewarded per security block according to a deterministic schedule (Section 6). Misbehavior is penalized through slashing: minor violations (downtime exceeding 100 blocks) incur a 5% stake penalty, while major violations (double signing) incur a 50% penalty. Slashed validators are jailed for 1,000 blocks (~50 minutes at 3-second commerce cadence).

6 Dual-Token Economics

6.1 Design Rationale

A single-token protocol forces one asset to serve simultaneously as a medium of exchange, a store of value, and a governance weight. These roles impose contradictory pressures: a medium of exchange must be abundant and low-friction; a store of value must be scarce; a governance weight must be illiquid enough to align long-term incentives. Omne separates these roles across two tokens—a design choice that, as formalized in Conjecture 1, we believe converges to a stable dual equilibrium under bounded network conditions.

6.2 OGT: Omne Governance Token

OGT is Omne's deflationary governance and staking token, governed by a fixed-supply halving schedule with hard cap.

Supply schedule. The initial block reward is 50 OGT per security block. The reward halves every 233,760 security blocks, an interval derived from the 9-minute block cadence:

$$233{,}760 \text{ blocks} \times 540 \text{ s/block} = 126{,}230{,}400 \text{ s} \approx 4.0 \text{ years}$$

The halving schedule proceeds through four reductions, after which a permanent floor of 3 OGT per block is maintained:

Epoch	Blocks	Reward (OGT)	Cumulative Supply (OGT)
0	0–233,759	50	11,688,000
1	233,760–467,519	25	17,532,000
2	467,520–701,279	12	20,333,120
3	701,280–935,039	6	21,735,680
4+	935,040+	3 (floor)	Asymptotic toward cap

The hard cap is 21,000,000 OGT (`OGT_MAX_SUPPLY = 21_000_000`). After the four halvings, the 3 OGT floor ensures continued validator compensation while maintaining supply within the cap through a diminishing issuance rate.

Staking and bonding. Validators must bond between 15 and 28 OGT. This dynamic range prevents both barrier-to-entry problems (stake too high) and Sybil attacks (stake too low). Delegation

is supported with a minimum of 1 OGT. Beyond validator staking, OGT serves as the bonding currency for protocol infrastructure operators: OON compute nodes bond 1,000 OGT and OMP storage nodes bond 500 OGT, creating a unified economic commitment model across all network roles.

6.3 OMC: Omne Commerce Token

OMC is the medium of exchange and fee currency. It is inflationary by design, with a fee recycling mechanism that maintains approximate supply neutrality. Unlike protocols such as Ethereum—where aggressive fee burning is desirable because ETH has no peg target and deflation increases token value—OMC targets a $1 soft peg. Permanent destruction of a large fraction of fees would create deflationary pressure that breaks this peg, making the commerce layer unpredictable for merchants and users who depend on price stability. Instead, 92% of transaction fees are recycled to validators as income via a dedicated `VALIDATOR_FEE_POOL_ACCOUNT`, while only 8% are permanently burned for anti-spam deterrence (`ANTI_SPAM_BURN_RATE = 0.08`).

Genesis supply. 1,000,000 OMC is minted at genesis to the OmneDAO treasury (`omne1000000000000000000000000`). There is no maximum supply.

Block reward. The initial OMC reward is 2 OMC per security block. Rewards follow the same halving schedule as OGT (233,760 blocks, 4 halvings):

Epoch	OMC Reward/Block
0	2.000
1	1.000
2	0.500
3	0.250
4+	0.125 (floor)

Precision. OMC uses 18-decimal precision:

$$1 \text{ OMC} = 10^{18} \text{ quar}$$
$$1 \text{ quar} = 10^{6} \text{ micro-quar}$$

The smallest unit of account is 1 micro-quar (10^{-24} OMC). This precision enables fee calculations at scales where rounding errors would otherwise dominate.

Treasury faucet and buy-back. The OmneDAO treasury operates a band peg mechanism for OMC price stability. The treasury sells OMC at a fixed ceiling price (faucet), providing a supply expansion valve when demand exceeds target. The treasury will buy back and burn OMC at a fixed floor price upon mainnet launch, providing a contraction valve when supply exceeds demand. The faucet (ceiling) side is operational on devnet; the buy-back (floor) side is a governance-activated mechanism that will engage with real market conditions on mainnet. Together, this mechanism bounds OMC volatility within a governance-defined band without requiring external collateral. In the language of Conjecture 1, conditions (3) and (4) are enforced by transaction volume caps and treasury solvency governance, respectively. Theorem 2 (Section 2.7) formalizes this mechanism

as a band-clamping controller and proves a quantitative lower bound on the peg's stability duration as a function of treasury reserves T_0 and demand volatility δ, with a critical threshold T_{min} that governance can monitor to trigger replenishment before the bound is approached.

7 Fee Economics

7.1 Design Principles

Omne's fee model is built on the premise that transaction fees should cover the marginal cost of execution, not the market-clearing price of block space. Fee revenue is explicitly not a protocol goal. The system cross-subsidizes low fees from three sources: OMC block rewards to validators, OON compute revenue (30% of which flows to fee subsidization), and fee recycling that routes 92% of transaction fees to validators as supplementary income while burning 8% for anti-spam deterrence.

7.2 Base Fee

The initial base fee is 20 micro-quar per unit of gas, dynamically adjusted based on block utilization with a target utilization of 50%:

$$\text{base\_fee} = 20 \text{ micro-quar/gas}, \quad \text{clamped to } [1, 100] \text{ micro-quar/gas}$$

In OMC terms, the base fee is:

$$20 \text{ micro-quar} = 20 \times 10^{-6} \text{ quar} = 20 \times 10^{-24} \text{ OMC}$$

At typical gas costs for a simple transfer (~21,000 gas), the fee is approximately 4.2×10^{-19} OMC—effectively zero in human-denominated terms.

Commerce discount. Transactions on the commerce layer receive a 50% fee reduction (`commerce_multiplier = 0.5`), reflecting the protocol's prioritization of commercial use cases.

7.3 Fee Recycling

OMC is a $1-pegged commerce token. This constraint fundamentally changes the relationship between fee mechanics and token economics compared to unpinned tokens like ETH. For Ethereum, aggressive EIP-1559 burning [9] is desirable: reducing supply increases token value, and there is no peg to break. For OMC, aggressive burning creates deflationary pressure that pushes the token above its $1 target, undermining the price stability that merchants and commerce applications depend on.

Omne resolves this tension through fee recycling. Transaction fees are split at a fixed ratio:

Destination	Share	Mechanism
Anti-spam burn	8% (`ANTI_SPAM_BURN_RATE = 0.08`)	Permanently destroyed — deters spam by ensuring attackers lose tokens irrecoverably
Validator recycling	92%	Credited to `VALIDATOR_FEE_POOL_ACCOUNT` — stays in circulation as validator income

The 8% burn rate is sufficient to make spam economically irrational—an attacker flooding the mempool permanently loses tokens—while small enough that normal commerce usage has negligible impact on circulating supply. The EIP-1559-style base fee *adjustment* mechanism still applies: the base fee increases when block utilization exceeds 50% and decreases below it, providing dynamic congestion pricing. What changed is the *destination* of collected fees: the vast majority are recycled rather than destroyed.

The net supply change per block is approximately:

$$\Delta_{\text{supply}} = R_{\text{block}} - (F_{\text{total}} \times 0.08)$$

where R_{block} is the block reward and F_{total} is total fees collected. Since both the block reward and the anti-spam burn are small relative to circulating supply, the net effect is approximate supply neutrality—precisely the property needed to maintain the $1 soft peg, and the empirical condition that Conjecture 1 predicts will hold under bounded transaction volume.

7.4 Revenue Cross-Subsidization

The Omne Orchestration Network generates revenue from off-chain compute jobs. A protocol-defined subsidy rate of 3,000 basis points (30%) of this revenue is routed to fee subsidization via PoVERA's `record_attested_job()` pathway. Revenue is smoothed over a rolling window of 12 security blocks (`REVENUE_SMOOTHING_WINDOW = 12`) to prevent fee volatility from compute demand spikes.

Storage fees collected by the Omne Media Protocol contribute to the same economic model. Escrow payments from asset uploaders flow to storage nodes over the lifetime of the stored assets, and the fee recycling mechanism applies to the on-chain component of storage manifest transactions. The combination of compute revenue (OON) and storage revenue (OMP) produces two independent, utilization-driven revenue streams that reinforce the system's ability to maintain microscopic commerce fees without subsidy expiration.

8 Deterministic Execution: The Axiom Runtime

8.1 Design Constraints

A commerce-grade blockchain requires that every validator executing the same transaction sequence arrives at the same state root, regardless of hardware architecture, operating system, or

compiler version. Nondeterministic execution is not merely a bug—it is a consensus failure. If two validators compute different state roots for the same block, the chain forks.

8.2 WASM Execution

The Axiom Runtime executes WebAssembly (WASM) bytecode in a sandboxed environment with deterministic memory allocation, floating-point behavior, and gas metering. WASM was chosen because it provides a well-specified, portable instruction set with no implicit platform dependencies. The runtime strips all sources of nondeterminism: system clocks, random number generators, and environment variables are replaced with deterministic host functions.

8.3 State Commitment

After each commerce block, the runtime produces a SHA-256 hash of the complete WASM memory state. At the security layer boundary, the `ExecutionStateCommitment` binds the state root, the execution trace hash, and the gas budget report into the security block:

$$\text{state\_root} = \text{SHA-256}\big(\text{state trie after 180 commerce blocks}\big)$$

Any validator can verify the commitment by replaying the commerce epoch and comparing the resulting hash. Disagreement triggers the slashing protocol.

Critically, the state root covers not only commerce block execution but also the protocol-level system operations (validator reward minting, fee distribution, fee vault seeding, and slash burns) that occur during security block production. Before the state root is read, the block producer snapshots all affected account states, executes the token operations via write-through persistence to RocksDB, diffs the pre/post states, and commits the resulting `StateChange` entries to the Merkle tree. This ensures that validators independently replaying the security block can reproduce the identical state root, and any discrepancy in reward or fee accounting is detectable through hash comparison.

8.4 Three-Tier VM

The Axiom Runtime provides three execution tiers, each with distinct resource budgets:

Tier	Timeout	Gas Limit	Memory	Block Budget
FastVM	400 ms	1,000,000	100 MB	70%
StandardVM	3 s	10,000,000	1 GB	20%
ComputeVM	30 s	100,000,000	4 GB	10%

Transactions are assigned to a tier at submission. Commerce-priority transactions execute in FastVM; general application logic in StandardVM; heavy computation (machine learning inference, cryptographic proofs) in ComputeVM. Tier budgets are enforced per-block, preventing compute-heavy workloads from crowding out commerce transactions.

9 Smart Contracts

9.1 Contract Deployment

Smart contracts are deployed via signed execution plans submitted through the `omne_deployContract` RPC endpoint. An execution plan contains the WASM bytecode, a designated entry function, the target VM tier, and deployment metadata. The contract address is derived deterministically:

$$\text{address} = \text{SHA-256}(\text{bytecode} \| \text{name} \| \text{entry\_function} \| \text{export} \| \text{nonce})$$

Deployed contracts are stored in the `ContractBytecodeRegistry` with the following fields:

Field	Description
`wasm_bytecode`	Compiled WASM binary (max 16 MB)
`bytecode_hash`	SHA-256 of the bytecode
`vm_tier`	Execution tier (FastVM, StandardVM, ComputeVM)
`entry_function`	Entry point function name
`export_name`	Exported WASM function identifier
`deploy_block`	Block height at deployment
`deployment_nonce`	Deduplication counter

9.2 Omne ABI

Omne uses a custom binary wire format for contract invocation rather than adopting existing account-model ABI conventions. Because the Axiom Runtime executes WASM natively, a purpose-built encoding that maps directly to WASM function signatures avoids the complexity of virtual machine translation layers designed for stack-based architectures. The wire format is:

```
[MAGIC: 4 bytes "OMNE" (0x4F4D4E45)]
[VERSION: 1 byte (0x01)]
[METHOD_NAME_LENGTH: 2 bytes, big-endian]
[METHOD_NAME: UTF-8 encoded string]
[ARG_COUNT: 2 bytes, big-endian]
[ARGUMENTS: type_tag (1 byte) + length (4 bytes) + data ...]
```

Nine argument types are supported:

Tag	Type	Size
0x01	U32	4 bytes
0x02	U64	8 bytes
0x03	I32	4 bytes
0x04	I64	8 bytes
0x05	String	Variable (length-prefixed)
0x06	Bytes	Variable (length-prefixed)
0x07	Bool	1 byte
0x08	Address	20 bytes
0x09	U128	16 bytes

Maximum calldata size is 4 MB. Maximum argument count per call is 64. Encoding and decoding are deterministic across all validators.

9.3 Contract Interaction

State-modifying calls. A transaction targeting a deployed contract address submits ABI-encoded calldata in the transaction's `data` field via `omne_sendTransaction`. The chain loads the contract's WASM bytecode from the registry, decodes the method name and arguments from the ABI envelope, and routes execution to the appropriate entry function within the Axiom Runtime.

Read-only queries. The `omne_call` RPC endpoint executes contract methods without committing state changes or consuming gas. This enables free balance lookups, metadata queries, and view functions.

Cross-contract calls. The `omne_host::call_contract` host function is registered as a WASM host import, establishing the entry point for composable multi-contract applications. Full nested execution with re-entrant state isolation is a planned upgrade; the current runtime returns an error status for cross-contract calls until the re-entrancy guard and call-depth budget are finalized.

9.4 SDK

The TypeScript SDK provides the `OmneContract` class for client-side contract interaction:

- `query(method, args)` executes a read-only call via `omne_call` and returns the decoded result.
- `call(options)` submits a state-modifying transaction and returns a `TransactionReceipt`.
- `AbiEncode` builders and `encodeContractCall()` produce correctly formatted ABI payloads.
- `isAbiEncoded()` validates that a byte array conforms to the Omne ABI wire format.

10 Omne Orchestration Network

10.1 Architecture

The Omne Orchestration Network (OON) is an off-chain distributed compute layer built directly into the protocol. Six modules—orchestrator, node registry, job distribution, coordination, VDP compliance, and attestation—are wired through a central `CoordinationManager`. Four RPC endpoints expose the system: `omne_submitJob`, `omne_getJobStatus`, `omne_registerNode`, and `omne_getNodeInfo`.

The design rationale is straightforward: the same validator infrastructure that secures the chain possesses compute capacity that sits idle between block production duties. Rather than leaving this capacity unused, OON allows validators (and third-party compute providers) to rent it for off-chain workloads, creating a revenue stream that directly subsidizes on-chain transaction fees.

10.2 Job Submission and Packaging

Jobs are submitted as self-describing `ComputeJobSubmission` envelopes containing:

- WASM bytecode (max 50 MB) with a SHA-256 integrity hash
- Input data (max 10 MB) with a SHA-256 integrity hash

- Entry function name
- Job type classification (Rendering3D, AiTraining, DataProcessing, Simulation, Cryptographic, or Custom)
- Resource hints (CPU cores, memory, GPU availability, storage, bandwidth)
- Maximum execution time and expected output size

Both the WASM bytecode hash and input data hash are verified at submission time. Any mismatch triggers deterministic rejection.

10.3 Chunked Execution

Long-running jobs execute in iterative 30-second time slices within the ComputeVM tier. Each chunk produces an `ExecutionCheckpoint` containing a memory snapshot that enables resumption after preemption or node reassignment. Budget enforcement operates across three dimensions:

Budget	Default Limit
Maximum chunks	4,320 (= 36 hours at 30 s/chunk)
Total gas	Configurable per job
Total wall time	Configurable per job

This model allows compute-intensive workloads—AI training runs, physics simulations, cryptographic proof generation—to execute within the protocol's deterministic framework without blocking commerce transactions.

10.4 Node Registry and Reputation

Compute nodes register with a bond of 1,000 OGT and declare their capabilities:

Capability	Description
`cpu_cores`	Available CPU cores
`memory_gb`	Available RAM
`gpu_available`	GPU compute support
`storage_gb`	Available storage
`bandwidth_mbps`	Network bandwidth
`vdp_compliant`	Supports Verifiable-Deterministic-Parallelizable framework

Nodes build reputation through job completion. The `ReputationHistory` track record factors reliability (completed vs. failed and timed-out jobs), performance, and quality into an overall reputation score. The minimum threshold for job assignment is 0.7 (`min_reputation_score = 0.7`). Nodes falling below this threshold are excluded from job distribution until their score recovers.

10.5 Dispatch and Verification

The `NodeDispatcher` routes sub-tasks to registered nodes via their endpoint URLs, enforcing a per-node concurrency cap of 10 concurrent jobs and a maximum payload size of 50 MB. Results are collected with timeout enforcement and retry logic (exponential backoff, max 5 attempts, 15-second initial / 900-second maximum backoff).

Every job result undergoes VDP compliance verification using one of four methods:

Method	Description
ZkProof	Zero-knowledge proof of correct execution
StateHash	SHA-256 hash comparison of output state
Replication	Independent re-execution on a second node
Benchmark	Performance comparison against known baselines

10.6 Attestation and Settlement

Completed and verified jobs produce cryptographic `ProofOfWork` certificates containing the work description, resource usage metrics, performance metrics, a computational proof, a VDP verification result, and an ed25519 signature. Certificates are valid for 24 hours.

Settlement occurs deterministically via the `SettlementManager`:

1. At job submission, the submitter's payment is locked in escrow.
2. Upon successful verification, a batch settlement transaction releases payment to the compute node. Transaction IDs are derived deterministically: `SHA-256("omne:oon:settlement:{batch_id}:{j`
3. If VDP verification fails post-settlement, a dispute mechanism claws back payment.

Compute revenue flows to fee subsidization at 3,000 BPS (30%) via PoVERA's `record_attested_job()` pathway. Consecutive job timeouts (default threshold: 3) trigger validator slashing through the PoVERA consensus mechanism.

11 Omne Media Protocol

Prior work in decentralized storage has produced protocols that operate as separate networks, each with its own token, consensus mechanism, and validator set. An application that requires both computation and storage must bridge between a smart contract chain and a storage chain, introducing latency, trust assumptions, and token conversion friction. We observe that this separation is an artifact of historical protocol design, not a fundamental constraint. Omne eliminates the boundary by building file storage directly into the protocol. Storage nodes participate in the same OGT staking and reputation infrastructure as validators and compute nodes, creating a unified economic model where compute, consensus, and storage are facets of a single network.

11.1 Architecture

OMP uses a hybrid on-chain / off-chain architecture. *Manifests* — asset metadata, SHA-256 content hashes, merkle roots, chunk assignments, and storage proof records — live on-chain in consensus state, benefiting from the security layer's finality guarantees. *Raw chunk data* lives

off-chain on dedicated storage nodes, avoiding the state bloat that would make on-chain storage prohibitively expensive at scale. The chain enforces data integrity; storage nodes provide data availability.

Nine coordinating modules — coordinator, storage registry, manifest manager, chunk distributor, chunk store, storage proofs, retrieval coordinator, settlement manager — are wired through a central `OmpCoordinator` that provides thread-safe access from the RPC layer and block execution pipeline. Six RPC endpoints expose the protocol: `omne_ompStoreAsset`, `omne_ompFinalizeAsset`, `omne_ompGetManifest`, `omne_ompRegisterStorageNode`, `omne_ompGetRetrievalPlan`, and `omne_ompStorageStats`.

11.2 Asset Lifecycle

The lifecycle of a stored file proceeds in four stages:

1. Manifest creation. The uploader submits a `storeAsset` request specifying the file's SHA-256 content hash, total size, chunk count, merkle root, redundancy level (2–7×, default 3×), storage tier (Hot, Warm, or Cold), and an optional erasure codec (Reed-Solomon or none). The protocol validates size constraints (maximum 1 GB per asset, `MAX_ASSET_SIZE = 1{,}073{,}741{,}824` bytes), redundancy bounds, and hash lengths, then creates an on-chain manifest in `Uploading` status and locks an OGT escrow for the estimated storage cost.

2. Chunk submission. The file is serialized into 256 KB chunks (`CHUNK_SIZE = 262{,}144` bytes, maximum 4,096 chunks per asset). Each chunk is content-addressed via SHA-256 and submitted to the chunk store with integrity verification on write. The chunk distributor assigns each chunk to `redundancy` distinct storage nodes using round-robin placement sorted by available capacity, ensuring uniform load distribution.

3. Finalization. Once all chunks are submitted, the uploader calls `finalizeAsset`. The protocol recomputes a binary SHA-256 merkle tree from the submitted chunk hashes and verifies the root against the declared merkle root in the manifest. A mismatch — indicating data corruption or tampering during upload — causes deterministic rejection. On success, the manifest transitions to `Finalized` status and becomes eligible for retrieval and proof-of-storage challenges.

4. Retrieval. Clients request a `RetrievalPlan` that maps each chunk index to an ordered list of storage node endpoints holding that chunk. Chunks can be fetched in parallel from multiple nodes. The client verifies each downloaded chunk against the SHA-256 hash from the on-chain manifest, providing end-to-end integrity without trusting any individual storage node.

11.3 Storage Nodes and Reputation

Storage nodes register by bonding a minimum of 500 OGT (`STORAGE_NODE_BOND_OGT = 500`) and declaring capabilities:

Capability	Description
`storage_gb`	Available storage capacity
`upload_bandwidth_mbps`	Upload throughput
`download_bandwidth_mbps`	Download throughput
`supports_hot_tier`	Immediate retrieval support

Capability	Description
`supports_cold_tier`	Archival storage support
`region`	Geographic region for latency optimization

The maximum storage node set is 10,000 nodes (`MAX_STORAGE_NODES = 10{,}000`). Each node begins with a reputation score of 1.0 and is recalculated based on proof-of-storage outcomes:

$$\text{reputation} = \frac{\text{proofs\_passed}}{\text{proofs\_passed} + \text{proofs\_failed}}$$

Nodes falling below a reputation threshold of 0.3 are automatically suspended and excluded from new chunk assignments. Only nodes with reputation ≥ 0.5 receive chunk distribution assignments. The reputation system creates a natural selection pressure: reliable nodes accumulate chunks and earn fees; unreliable nodes lose assignments and eventually forfeit their bond.

11.4 Proof of Storage

Validators periodically challenge storage nodes to prove they still hold specific chunks. The challenge-response protocol operates as follows:

1. The challenger (a validator) generates a random nonce and constructs a `StorageChallenge` with a deterministic challenge ID:

$$\text{challenge\_id} = \text{SHA-256}(\text{node\_id} \| \text{asset\_id} \| \text{chunk\_index} \| \text{nonce} \| \text{timestamp})$$

2. The challenged storage node computes the expected proof response:

$$\text{proof} = \text{SHA-256}(\text{chunk\_hash} \| \text{nonce})$$

3. The validator verifies the response by recomputing the same hash from the on-chain chunk hash (known from the manifest) and the nonce it issued. Three outcomes are possible:

Verdict	Condition	Consequence
Valid	Proof matches expected hash	Proofs passed counter incremented
Invalid	Proof does not match	Reputation penalty; consecutive failures trigger suspension
Expired	Response received after deadline	Treated as failure; reputation penalty

The default challenge interval is 3,600 seconds (`PROOF_CHALLENGE_INTERVAL_SECS = 3{,}600`). Hot-tier assets are challenged more frequently than cold-tier, reflecting the higher availability requirements.

11.5 Storage Tiers

OMP supports three storage tiers, each with distinct performance and cost characteristics:

Tier	Retrieval Latency	Cost	Proof Frequency	Use Case
Hot	Immediate	Highest	Most frequent	Real-time media, application assets
Warm	Moderate	Balanced	Standard	Documents, historical data
Cold	High	Lowest	Least frequent	Backups, archival, compliance records

Tier selection is made at manifest creation and affects pricing, proof challenge frequency, and the storage node eligibility filter (only nodes declaring tier support receive assignments for that tier).

11.6 Settlement

Storage fees use an escrow-based periodic drip model. When an asset manifest is created, the uploader's OGT is locked in escrow. Over the storage period, escrow funds are released incrementally to the storage nodes holding the asset's chunks. Settlement IDs are derived deterministically:

$$\text{settlement\_id} = \text{SHA-256}\big(\texttt{"omne:omp:settlement:\{asset\}:\{node\}:\{block\}"}\big)$$

This deterministic derivation ensures that any validator can independently reconstruct the settlement history from the chain state. When the escrow is fully released, the settlement record is marked complete. If a storage node fails proof-of-storage challenges, the corresponding portion of the escrow is slashed rather than released, aligning storage node incentives with data availability.

11.7 Design Rationale

Building storage into the protocol rather than delegating it to a separate network eliminates three classes of problems:

1. **Bridge risk.** External storage networks require bridge contracts, oracles, or multi-sig committees to relay storage proofs to the application chain. Each bridge is an attack surface. OMP's proofs are verified natively by the same validators that produce blocks.
2. **Token friction.** Separate storage networks use separate tokens. Users must acquire, manage, and convert between tokens to use storage and compute together. OMP uses OGT for bonding and OMC for fees — the same tokens used for everything else on the network.
3. **Incentive fragmentation.** When storage and compute are on different networks, there is no mechanism for cross-subsidization. OMP storage fees contribute to the same economic

system that keeps commerce fees microscopic: storage manifest transactions contribute to fee recycling (8% anti-spam burn, 92% validator recycling), and storage node bonds contribute to the OGT staking economy.

12 Security Model

12.1 Anti-Spam Controls

Omne does not rely on fee escalation to deter spam. Instead, the protocol enforces layered non-fee controls:

Mempool rate limiting. Transactions are rate-limited at three granularities:

Scope	Rate	Burst
Per account	60 tx/s	20 tx burst
Per source IP	300 tx/s	75 tx burst
Global	8,000 tx/s	1,000 tx burst

The mempool is capped at 10,000 pending transactions with a maximum payload of 512 KB per transaction.

Stake-weighted QoS. Validators with higher stake receive proportionally higher priority in block production and message propagation, ensuring that well-capitalized honest validators can always outcompete an attacker with equal or lesser stake.

Circuit breakers. Under sustained overload, the protocol activates circuit breakers that shed low-priority traffic while preserving consensus and security message throughput.

12.2 Slashing

Violation	Penalty	Jail Duration
Minor (downtime > 100 blocks)	5% of stake	1,000 blocks (~50 min)
Major (double signing)	50% of stake	1,000 blocks (~50 min)

Slashed OGT is burned and tracked in the security block's `slashed_stake_ogt` field, providing a public audit trail.

12.3 Cryptography

Omne standardizes on Ed25519 for all signature operations across the protocol: transaction signing, block signing, validator identity, OON attestation certificates, OMP storage proof verification, and cross-contract message authentication. Ed25519 was chosen for its performance characteristics (batch verification), resistance to side-channel attacks, and absence of the nonce-reuse vulnerabilities that affect elliptic-curve schemes requiring per-signature randomness.

Block signing. Each validator derives a deterministic Ed25519 consensus signing key from its P2P identity material at startup. Security blocks are signed after hash computation: the block hash (which covers the PoVERA data, commerce commitment, and execution state commitment) is signed with the validator's consensus key, and the 64-byte Ed25519 signature plus 32-byte public key are embedded in the block's `proposer_signature` and `proposer_public_key` fields. Receiving nodes recalculate the block hash from embedded fields, verify the Ed25519 signature, and confirm that the signing key matches the consensus key registered for the proposer address in the validator registry. Invalid signatures trigger deterministic block rejection.

BFT vote verification. Before casting a BFT attestation vote, each receiving validator independently re-verifies the block's system operations against six accounting invariants: vault seed balance consistency, fee distribution totals, OGT and OMC mint totals, absence of zero-amount operations, and validator reward coverage. This ensures that no security block can achieve BFT finality without independent economic verification by a supermajority of validators.

12.4 Address Format

All on-chain addresses use the `omne1` prefix followed by a 40-character hex encoding of the 20-byte address. The `omne1` prefix serves as a human-readable network identifier, distinguishing Omne addresses from those of other networks at a glance and preventing cross-chain address confusion. The genesis configuration, account parsers, and RPC layer enforce this format uniformly.

13 Governance

The Omne protocol is governed by OmneDAO, a Swiss Foundation. Protocol parameters—fee multipliers, burn rates, stake ranges, slashing penalties, OON subsidy rates—are adjustable through OGT-weighted governance proposals. The DAO treasury, seeded with 1,000,000 OMC at genesis, funds protocol development, ecosystem grants, and the OMC band peg mechanism.

OGT holders participate in governance with voting power equal to their balance plus any locked (staked) tokens. This ensures that validators—who have the most direct operational interest in the protocol's health—hold proportional governance weight.

14 Ecosystem: The Three-Tier Token System

Omne supports a three-tier token architecture:

1. **OMC (Infrastructure).** The base-layer medium of exchange and fee currency. Used for all on-chain transactions, smart contract interactions, OON compute payments, and OMP storage manifest fees.
2. **OGT (Governance).** The staking, slashing, and governance token. Used for validator bonding (15–28 OGT), OON compute node registration (1,000 OGT), OMP storage node registration (500 OGT), and protocol-level voting.
3. **ORC-20 (Application).** A token standard for application-specific utility tokens deployed as smart contracts. ORC-20 tokens inherit OMC-denominated gas costs automatically. The standard defines:

- Core operations: `balance_of`, `transfer`, `approve`, `transferFrom`
- Governance integration: `lock_tokens`, `unlock_tokens`, `voting_power`
- Configuration: maximum supply, mintability, burnability, transfer controls (blacklists, per-transaction limits, daily volume caps)
- Fee sharing: configurable platform revenue sharing and transfer fee rates

This layered design enables application developers to issue tokens with custom economics while inheriting Omne's fee infrastructure and security guarantees.

15 Observations and Conclusions

The architecture described in this paper yields several observable properties that address the claims stated in Sections 1 and 2.

Observation 1: Cadence separation resolves the latency-security trade-off. The CSC framework (Section 2) predicts that dual-cadence block production achieves commerce-grade latency without compromising settlement finality, with security overhead amortized at $O(1/k)$. Omne's instantiation with $k = 180$ confirms this prediction: the 3-second commerce layer provides point-of-sale confirmation, the 9-minute security layer provides BFT settlement guarantees, and the 180:1 rollup ratio binds the two layers through SHA-256 commitments that any validator can independently verify. The Binding Theorem (Theorem 1) provides the formal basis: commerce transactions achieve probabilistic finality at τ_c and settlement finality at τ_s, with the security cost of the BFT protocol amortized across 180 commerce blocks per round.

Observation 2: Dual-token economics support the Dual Equilibrium Conjecture. By separating the medium of exchange (OMC) from the governance and staking instrument (OGT), the protocol avoids the contradictory pressures that force single-token designs to choose between scarcity, liquidity, and governance alignment. OGT follows a deflationary halving schedule with its liquid supply decreasing as staking and infrastructure bonding demand absorbs issuance; OMC is inflationary with fee recycling that maintains approximate supply neutrality, preserving its $1 soft peg. Neither token is asked to serve a role that conflicts with its supply mechanics. These dynamics match the equilibrium conditions stated in Conjecture 1: OGT's liquid supply decreases monotonically toward a staking-determined minimum, OMC's circulating supply oscillates within a bounded range, and the net supply change per block converges toward zero as the network matures. The Peg Stability Theorem (Theorem 2) strengthens this observation: the treasury's band-clamping controller maintains OMC within $[p_f, p_c]$ for at least $t^* = \lfloor T_0/(\delta \cdot \Delta p_{\text{max}}) \rfloor$ blocks, providing a quantitative and falsifiable bound on peg duration as a function of treasury reserves and demand volatility. The Layered Duality principle (Design Principle 1) explains *why* this pairing is structurally natural rather than ad hoc: each token's supply dynamics are coupled to the cadence layer it serves.

Observation 3: Microscopic fees are sustainable as a structural property. Transaction fees at the cost floor—on the order of 10^{-20} OMC per simple transfer—are maintained not through temporary subsidy but through three reinforcing mechanisms: OMC block rewards to validators, fee recycling that routes 92% of transaction fees to validators as supplementary income (with 8% burned for anti-spam deterrence), and revenue cross-subsidization from the Omne Orchestration Network (30% of compute revenue) and the Omne Media Protocol (storage fee settlement). The combination of two independent, utilization-driven revenue streams reduces the risk that any single demand source failing would compromise fee sustainability. Conjecture 2 (Structural Fee

Sustainability) formalizes this claim: under the sustainability condition $R_M^{\min} + \gamma \cdot \Omega_{\min} > n \cdot c_v$, validator revenue exceeds operating cost independent of fee volume, making microscopic fees a permanent architectural property. This structural approach to fee economics is a direct consequence of the CSC framework: because the security layer amortizes BFT costs across k commerce transactions, the per-transaction security overhead is $O(1/k)$, allowing fee levels to be set by execution cost rather than consensus cost.

Observation 4: Protocol-integrated compute and storage eliminate bridge risk. The Omne Orchestration Network and the Omne Media Protocol demonstrate that off-chain compute and decentralized storage can share a single staking, reputation, and settlement infrastructure with the consensus layer. This eliminates the token conversion friction, bridge trust assumptions, and incentive fragmentation that arise when these services operate as separate networks. Smart contracts execute deterministically across heterogeneous hardware, files are stored with cryptographic proof of availability, and off-chain compute workloads settle on-chain with verifiable attestation—all within a unified economic model. The orchestration, node registry, job distribution, storage proof, and settlement modules are implemented and tested; live compute marketplace and binary chunk data-plane operations will activate alongside the testnet validator community.

Observation 5: The CSC framework generalizes beyond Omne. The theoretical contributions of this paper—the Binding Theorem, the Dual Equilibrium Conjecture, the Peg Stability Theorem, the Structural Fee Sustainability Conjecture, and the Separability Property—are not specific to Omne's parameter choices. The Binding Theorem holds for any cadence ratio $k \geq 2$ and any BFT-capable security-layer protocol satisfying the $\lfloor 2n/3 \rfloor + 1$ quorum condition. The Dual Equilibrium Conjecture applies to any dual-token system satisfying conditions (1)–(4), regardless of the underlying consensus mechanism. The Peg Stability Theorem applies to any band-clamping treasury mechanism with bounded demand volatility. The Separability Property (Property 1) decomposes verification into three independent obligations for any CSC instantiation. We have described one point in the CSC design space; the space itself remains largely unexplored. A high-frequency trading chain, a settlement network, and a multi-cadence archival system would each make different parameter choices while inheriting the same formal guarantees. We invite the research community to investigate alternative instantiations, to attempt formal proofs or counterexamples for Conjectures 1 and 2, and to extend the framework to multi-cadence systems with three or more layers.

These findings support the hypothesis that the latency-security-fee trilemma is resolvable through cadence separation. The design space explored here—cadence-separated consensus, dual-token economics, non-fee anti-spam, integrated compute, and integrated storage—represents one feasible point in that space. We invite independent analysis, formal verification of the Binding Theorem's assumptions, adversarial testing of the security model, economic simulation of the Dual Equilibrium Conjecture's boundary conditions, and empirical validation of the Peg Stability Theorem's quantitative bounds [11, 12].

References

[1] A. Kiayias, A. Russell, B. David, R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol," Crypto 2017.

[2] E. Buchman, J. Kwon, Z. Milosevic, "The Latest Gossip on BFT Consensus," 2018.

[3] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang, “High-Speed High-Security Signatures,” Journal of Cryptographic Engineering, 2012.

[4] A. Haas et al., “Bringing the Web up to Speed with WebAssembly,” PLDI 2017.

[5] R. C. Merkle, “Protocols for Public Key Cryptosystems,” IEEE Symposium on Security and Privacy, 1980.

[6] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, I. Abraham, “HotStuff: BFT Consensus with Linearity and Responsiveness,” ACM PODC, 2019.

[7] L. Lamport, “Time, Clocks, and the Ordering of Events in a Distributed System,” Communications of the ACM, 21(7), 1978.

[8] M. Castro, B. Liskov, “Practical Byzantine Fault Tolerance,” OSDI, 1999.

[9] T. Roughgarden, “Transaction Fee Mechanism Design,” ACM SIGecom Exchanges, 2021.

[10] NIST, “Secure Hash Standard (SHS),” FIPS PUB 180-4, 2015.

[11] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice Hall, 2002.

[12] E. D. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” SIAM J. Control Optim., 21(3), 1983.